

AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions of claims in the application:

Listing of Claims:

1. (Currently Amended) A computer readable medium having computer executable program code embodied thereon for performing the following act for authoring of rules:

composing a rule based on one or more statements, such that the rule is decomposed into a subset of instructions, the instructions are processed asynchronously to control utilization of computer resources and to facilitate at least one of testing assertions, enforcing constraints using runtime information, making inferences, performing correlation, or communicating results of dynamic tests to other components, the rule executed concurrently with one or more disparate rules in an infinite loop with suitable timeouts to keep the rule from running all the time, the rule associated with a rule type is defined by RDL (rules definition language) code while a separate XML document is employed to create instances of the rule type by specifying the parameter values necessary for its instantiation;

mapping a thread to at least one task for execution and employing the thread to execute multiple tasks concurrently;

providing primitives for automatically correlating asynchronous observations from a plurality of sources to eliminate the need to write multithreading code; and

determining a model that is utilized for a given instance or implementation by employing an artificial intelligence scheme.

2. (Previously Presented) The computer readable medium of claim 1, at least one of the one or more statements facilitates activating the rule for processing.

3. (Previously Presented) The computer readable medium of claim 1, at least one of the one or more statements facilitates activating the rule according to least one of periodically or on detection of an event.

4. (Previously Presented) The computer readable medium of claim 1, the one or more statements facilitates performing a continuous polling loop.
5. (Previously Presented) The computer readable medium of claim 4, the polling loop is performed according to a polling interval such that the polling loop executes and then waits for the polling interval to expire before executing a next polling loop.
6. (Previously Presented) The computer readable medium of claim 4, the polling loop is facilitated according to a keyword that includes at least one of a variable, an object, or a constant.
7. (Previously Presented) The computer readable medium of claim 1, the rule executes concurrently with another rule.
8. (Previously Presented) The computer readable medium of claim 1, at least one of the one or more statements facilitates creating at least one of an implicit task or an explicit task, wherein the explicit task is created by explicitly specifying a keyword and an explicit task object, and the implicit task is created when a startup attribute is used on the rule.
9. (Previously Presented) The computer readable medium of claim 8, the statements facilitates explicit task declaration of the explicit task object for the explicit task, and use of a statement that launches concurrent execution of the rule.
10. (Previously Presented) The computer readable medium of claim 1, the one or more statements facilitates creating a task using one of a startup attribute or a signaling attribute.
11. (Previously Presented) The computer readable medium of claim 1, the one or more statements facilitates allowing the rule to invoke another rule.
12. (Previously Presented) The computer readable medium of claim 1, the one or more statements facilitates separating rule logic data from rule configuration data using at least one parameter.

13. (Previously Presented) The computer readable medium of claim 12, the at least one parameter is passed by one of a value or by reference.
14. (Previously Presented) The computer readable medium of claim 1, the rule is an independent rule authored using at least one of an infinite loop or an event-driven callback.
15. (Previously Presented) The computer readable medium of claim 14, the event-driven callback facilitates asynchronous delivery of a data item from a URI (uniform resource identifier) -based source.
16. (Previously Presented) The computer readable medium of claim 1, the rule is subscribed to reveal events at one time.
17. (Previously Presented) The computer readable medium of claim 1, the one or more statements facilitates collecting at least two data items concurrently, when the at least two data items become available.
18. (Original) The computer readable medium of claim 1, embodied within a device.
19. (Previously Presented) The computer readable medium of claim 1, the one or more statements facilitates at least one of automated rule instantiation based on XML (Extensible Markup Language), built-in polling without threading or concurrency considerations, and automated logging of rule instance information in alerts.
20. (Previously Presented) A computer readable medium having computer executable program code embodied thereon for providing a method that performs the act of authoring rules for concurrent processing, the method comprising:
composing a rule of one or more statements that facilitate decomposing the rule into a subset of instructions in accordance with the one or more statements, the instructions are processed at least one of asynchronously or out-of-order to improve utilization of computer resources, the rules processed to perform at least one of testing assertions, enforcing constraints

using runtime information, making inferences, performing correlation, or communicating results of dynamic tests to other components, the rule associated with a rule type is defined by RDL (rules definition language) code the while a separate XML document is employed to create instances of the rule type by specifying the parameter values necessary for its instantiation;

mapping at least one thread to one or more tasks for execution and employing the at least one thread to execute several tasks concurrently;

providing primitives for automatically correlating asynchronous observations from a plurality of sources to eliminate the need to write multithreading code; and

determining one or more models that are utilized for a given instance or implementation by employing an artificial intelligence scheme.

21. (Previously Presented) The method of claim 20, further comprising extension of the rule with constructs without modifying the rule, the extension is performed by one of,

extending the rule to allow an alternate test before allowing the rule to fail;

constraining the rule to make the test more stringent; or

hooking the rule.

22. (Original) The method of claim 20, further comprising forwarding a log event to a supervisor in accordance with the one or more statements of the rule.

23. (Original) The method of claim 20, further comprising forwarding a log event using a function in accordance with the one or more statements of the rule, the log event forwarded to a supervisor that deployed the rule from which the function is called.

24. (Original) The method of claim 20, further comprising forwarding an alert using a function in accordance with the one or more statements of the rule, the alert forwarded to a supervisor that deployed the rule from which the function is called.

25. (Original) The method of claim 20, further comprising,
consolidating a plurality of events; and
reporting a consolidated event summary based on the consolidated events according to a
predetermined time interval.
26. (Original) The method of claim 20, further comprising monitoring system hardware and
software resources in accordance with the one or more statements of the rule.
27. (Currently Amended) A computer readable medium having computer executable
program code embodied thereon for providing a method of authoring rules for concurrent
processing, the method comprising[. . .]:

employing a rules definition language (RDL) to compose a rule of one or more
statements that facilitate decomposing the rule into a subset of instructions that control utilization
of computer resources by independent scheduling for out-of order execution, the logic of the rule
representative of an infinite loop, the rules processed to perform at least one of testing assertions,
enforcing constraints using runtime information, making inferences, performing correlation, or
communicating results of dynamic tests to other components, the rule associated with a rule type
is defined by RDL (rules definition language) code while a disparate XML document is
employed to create instances of the rule type by specifying parameter values necessary for its
instantiation;
mapping at least one thread to one or more tasks for execution;
employing the at least one thread to execute several tasks concurrently;
automatically correlating asynchronous observations from a plurality of sources by
providing one or more primitives to eliminate the need to write multithreading code; and
employing an artificial intelligence scheme to determine one or more models that are
utilized for an instance or implementation.